Atty. Dkt. No. 03CR254/KE (047141-0348)

<u>**Amendments to the Specification:**</u>

Please amend the specification as follows:

Please replace paragraph numbers [0002], [0003], [0004], [0005], [0006], [0018], [0019], and [0020], with the following respective rewritten paragraphs:

[0002]   Communication middleware resides between client and server applications in distributed systems. Middleware simplifies application development by providing a uniform view of networks, protocols, and operating system layers. Object Request Brokers (ORBs), such as ~~COBRA~~ <u>Common Object Request Broker Architecture (CORBA)</u> , DCOM, and Java RMI, eliminate many of the tedious, error-prone, and non-portable aspects of developing and maintaining applications programmed using mechanisms like sockets. In particular, ORBs automate common network programming tasks, such as object location, object activation, parameter (de)marshalling, socket and request demultiplexing, fault recovery, and security.

[0003]   The United States Department of Defense has mandated compliance to the Joint Tactical Radio System (JTRS) Software Computer Architecture (SCA) for all new communication radios that operate between 2 MHz and 2000 MHz. The SCA further requires software applications to run under an operating system and that data be passed using ~~COBRA~~ <u>CORBA</u> middleware. CORBA (Common Object Request Broker Architecture) provides platform-independent programming interfaces and models for portable distributed-oriented computing applications.

[0004]   Conventional CORBA implementations have limitations that make it hard for them to support performance-sensitive applications effectively. For example, conventional ORBs like ~~COBRA~~ <u>CORBA</u> support a limited number of statically configured protocols, often just GIOP (General Inter-ORB Protocol) / IIOP (Internet Inter-ORB Protocol) over TCP/IP. Further, conventional ORBs do not allow applications to configure key protocol policies and properties, such as peak virtual circuit bandwidth or cell pacing rate. Moreover, conventional ORBs do not support simultaneous use of multiple inter-ORB messaging or transport protocols.

-2-

Yet further, conventional ORBs have limited or no support for specifying and enforcing real-time protocol requirements across a backplane, network, or Internet end-to-end.

[0005] Another significant disadvantage of commercial middleware, such as ~~COBRA~~ CORBA, and operating systems available today is they are much too slow to allow use in power-constrained applications, such as the Objective Force Warrior, munitions and Unattended Ground Sensors (JTRS Cluster "5"). By adding sufficient processing power to overcome the latency caused by the operating system and middleware, the resulting heat and battery life reductions make compliance with JTRS SCA requirements virtually impossible. For non-battery powered applications, such as JTRS Cluster 1, there is a tremendous recurring cost and heat burden that the program cannot reach. A large part of the performance demand is required performance because of the excessive heat dissipated by numerous 1000 MIP processors.

[0006] One attempt to resolve some of the limitations of middleware, such as ~~COBRA~~ CORBA, is implementing a pluggable protocol framework. A pluggable protocol framework refers to using customized ORB interfaces that allow data to be passed without going through the operating system. The pluggable protocols are software applications designed using a standard protocol configuration to allow custom messaging and transport protocols to be configured flexibly. Nevertheless, software pluggable protocols are similar in speed to the middleware application and they may not be portable across platforms.

[0018] Fig. 2 illustrates a functional view of an architecture for power restricted applications according to known systems. A waveform application 22 communicates with a waveform application [[40]] 42 using a variety of functions.

[0019] Conventional systems utilize a user data protocol (UDP) / transmission control protocol (TCP) interface 28 or a proprietary connections interface 30. The UDP / TCP interface 28 communicates data to an Ethernet 34 and a SCMP (Snuggly Coupled Multi Processor) interface 36. In like fashion, the ORB [[42]] 44 provides data to a user data protocol (UDP) /

-3-

transmission control protocol (TCP) interface 48 and a proprietary connections interface 50. The UDP / TCP interface 28 communicates data to an Ethernet 54 and a SCMP 56.

[0020]   Fig. 3 illustrates a functional view of an architecture for power restricted applications included in the communication apparatus 10 of Fig. 1. A waveform application 22 communicates with a waveform application [[40]] 42 using a variety of functions. Additional, fewer, or different functions may be performed in the communication process, depending on the particular embodiment.

MADI_987868.1